Attack Trees and Their Uses in BGP and SMTP Analysis

Charles Marshall
Department of Computer Science
Drexel University
Philadelphia, PA 19104
cmm77@drexel.edu

Abstract

Recently, there have been attacks against several privacy data warehouses, databases which hold the personal information of thousands of people. This data has been released into the public, and many identity theft claims have been made as a result. This leads to confusion, distrust, and frustration on the part of those attacked. The question which naturally results is, “Why wasn’t my data protected against this attack?”

Often, flaws in security are overlooked in networks, either through ignorance of the types of threats, simple neglect, a lack of resources, or a combination of all three. Usually, it is the lack of resources which leads to the infiltration of a computer network, and this led to disastrous results in the past, such as the identity theft situation described above. Lack of resources can play a large role in the insufficient security of a computer resource, such as a network. There are various ways of tackling this problem, but it all starts with knowing what the potential attack is, and then trying to develop a countermeasure to that attack. Knowing where to place the effort in protecting against a potential threat is of critical importance when one’s defense resources are limited. Enumeration of the different types of attacks and the probabilities of those attacks can therefore be used to predict where and how often a type of attack will occur. One method of listing out these attacks is what is known as ‘attack trees’.

This paper describes and defines attack trees, what they are, how they are used, and why they have been instrumental in developing plans for protecting against attacks in two types of protocols, the Border Gateway Protocol and the Simple Mail Transfer Protocol.

Categories and Subject Descriptors

Security analysis, network protection, attack trees, protocols

1. Introduction

Security is a term that perhaps gets overused in recent years. Security in software applications is the process by which one protects certain resources from others. Software developers have not quite arrived at the crux of the problem: some software is inherently insecure. Some software is insecure due to poor design; some software is insecure due to flaws in the logic the developer uses in her algorithms. But the problem of some software lies in the insecure foundations on which the program has been built. If the tools which comprise the application are not assured to be completely secure, then there exists a possibility for someone to exploit the weaknesses of the insecure software. This paper will examine attack trees, a tool for finding, from the attacker's point of view, the most efficient means of exploiting potential weaknesses within software. This paper will accomplish this by providing an attack tree which has been used to spot flaws in a protocol, the Border Gateway Protocol (BGP) and examining the means through which an attack can cause damage to a system which uses it. Additionally, this paper will provide a potential attack tree for another protocol, the Simple Mail Transfer Protocol (SMTP), and show that this can be used as an effective tool for both those wishing to cause harm to a system and for those who wish to shore up their system security.

2. Attack Trees

Any potential attacker has a goal, be it burgling a house, opening a safe, or exploiting a weakness in the structure of a software protocol; this is the 'end'.
'Attacking' a protocol is the means to get to an end, which in this case is a computing resource, which the attacker wants. If the attacker is a rational person, which is what we assume for the purposes of this paper, then the attacker will want to expend the least amount of their resources in achieving that end. This minimizes the amount of time/energy/money expended and maximizes the potential harm they can cause.

An attack tree is either a graphical or textual representation of the means to and end. See Figure 1 and 8 of [1] for the general layout of an attack tree. The layout of the 'tree' starts with a root node, and has one or more branches to other nodes. Typically, the end is placed at the root of the tree; this is the attacker's goal. This goal is achieved through some means, which can be represented as child nodes. This leads directly to a tree structure; the means to get to a goal can be thought of as an end itself. If one wants to achieve this end, one must first accomplish the sub goals for this end. There are three types of nodes associated with an attack tree: OR nodes, AND nodes, and Leaf nodes. The first two are self-explanatory, but will be addressed here.

The first type of attack tree node is an OR node. Some goals may not require all of its sub goals to be accomplished in order for it to be considered completed. For example, these goals can be achieved by accomplishing any one of its sub goals. This is represented in the attack tree as an OR node. In comparison to its Boolean logic equivalent, if any of the sub goals are achieved, then the goal is considered accomplished. In addition, if all of its sub goals are not achieved, then the goal is considered not accomplished.

The second type of attack tree node is an AND node. Like the OR node, this is comparable to its Boolean logic equivalent in the sense that if all of its sub goals are achieved, this goal is considered accomplished. Likewise, if any one of its sub goals is not achieved, then this goal is considered not accomplished.

The last type of attack tree node is the Leaf node. This is a task which cannot be further subdivided into smaller goals. This type of task is considered an atomic event, and will be executed first in the attack plan. This makes sense, since any goal must first achieve its sub goals before deciding whether or not this goal can be considered accomplished.

While this definition is sufficient to explain simple attack trees, more complex and relevant trees for the purposes here, also have costs associated with specific actions, probabilities of success, and some measure of the ability to achieve that task. In addition, there must also be some measure of what the potential impact of achieving that goal is, i.e. the cost to the person protecting the resource. Most often this is expressed in terms of money, but this can also be expressed as time spent recovering from the attack, the level of effort it would take to recover, etc.

3. BGP Attack Trees

There has been much research done with regards to the potential attack points in the Border Gateway Protocol. The Border Gateway Protocol is the method used by gateways of an autonomous system to communicate with one another. An autonomous system is a set of networks, often under a single domain, which use a common routing policy. Typically, BGP is used as a means to communicate between different Internet Service Providers. Constructing BGP attack trees can therefore show ISPs what vulnerabilities are most hazardous to their business and which attacks are most likely to occur, both from inside and outside the network systems.

There are several different attack trees which can be constructed to illustrate the potential attack points of the Border Gateway Protocol (BGP). A useful, although simple tree for resetting a single BGP session may look like [5, page 7]. This attack tree shows that has two choices to reset a session: one may either alter the configuration of an already compromised router, or send a message to a router (either compromised or not) causing the reset of the router. The first option is almost certainly not probable. The attacker is almost certainly executing the attack from the "outside" of the system, and therefore will not have direct access to the configuration of the router. The administrator of the system may want to keep this in mind, but he/she will almost certainly want to focus more effort on the other path of the tree, the message sent to the router, which causes the reset. Notice that in the third level of the attack tree, there are three tasks which can be accomplished, but at the very least the TCP Sequence number attack must be accomplished, and either of the other two tasks, "Send RST message to TCP stack" or "Send BGP Message", must also be accomplished. The TCP Sequence number Attack is a method of hijacking a TCP communication by sending false TCP packets, which purport to come from the trusted source.

At first glance, it would seem that the route of "Send RST message to TCP stack" would be the best way to attack, since this along with the requisite "TCP Sequence number Attack" accomplishes the task of "Send message to router causing reset", which in itself accomplishes the root goal. This attack tree, however, does not include any mention of costs to use a specific attack method, the difficulty of the attack, the skill of the attacker, or any of a number of other factors which indicate how likely that method of attack is.

4. Another BGP Attack Tree

More in depth, if not more expressive attack trees, can be seen in textual form in [7, pages 6-10]. Among the list of goals for this attack tree are:

- Compromise MD5 authentication

- Establish unauthorized BGP session with peer
- Originate unauthorized prefix into peer route table
- Change path preference of a prefix
- Conduct denial/degradation of service against BGP process
- Reset single BGP session
- Spoof a BGP message

Notice that the attack trees listed in [7], like [5], do not mention anything about level of difficulty, so some research must be done by the administrator to determine where the effort must be expended in order to adequately protect the system. This in itself is a matter which the system/network administrator must discuss with the security experts to determine the best level of resource expenditure.

1. As [7] states, the most likely path a potential attack will take is the path of least resistance. This means that taking the first path of compromising the MD5 password is not the most likely route an attacker would take to compromise the system. In the instance of compromising the MD5 password, as seen in [7], this would take either using social engineering to obtain the password, capturing the password, brute-force attacking the MD5 password, discovering an implementation flaw in the MD5 authentication, discovering a new attack against MD5, or exploiting the hash collision attack against MD5 [7]. Obviously, attacking using either of the first two attacks is the easiest approach. A brute-force attack would be futile, because the search space for the key is 2^128 bits for MD5; this is simply enormous, and it would take far too long to guess the key. Likewise, discovering a flaw in the implementation of the RFC 2385 standard and discovering a new attack against MD5 and exploiting the collision attack against MD5 are all extremely unlikely.

2. The next attack, establishing and unauthorized BGP session with a peer is essentially establishing a peering arrangement with both sides of a session without the permission of both sides of the peer session. A peering arrangement is defined as an agreement between BGP routers to exchange traffic. This type of attack has fewer sub goals to achieve in order to accomplish this goal. This attack, however, requires a good deal of social engineering on the part of the attacker, or a good port scanner to determine the local BGP speaker and the remote BGP speaker for the first attack, while the second attack requires physical access or local administrative access to the BGP routers. This is also not a likely attack, due to the risk involved. The only real potential attack from this attack group is the third type of attack, which is to simulate a BGP session from a non-router. This, however, requires a great amount of skill on the part of the attacker, and is therefore not as likely as other attacks. The payoff would be extremely high, though, for the attacker, as he/she could use that session to sniff private data traveling between the two autonomous systems.

3. Originating an unauthorized prefix into the peer route table is another possible attack against the protocol, and this one requires the router to be either configured

incorrectly to start with, or stopping the originating router and introducing a new 'rogue' router, or sending a spoofed BGP message via a sequence number attack or a man-in-the-middle attack. Realistically, for physically secured a router, which is what this paper assumes, the only option would be the third attack, the spoofed BGP Update from a non-router. Obviously, this will still require a level of skill and knowledge. However, it is good to know that physical security is still a good means of preventing a percentage of attacks against this protocol.

4. Another type of attack is the changing the path preference of a prefix. Looking at the attack tree, there are only three ways of accomplishing this task; either have a rogue transit implementation or compromise the edge router of a valid BGP announcer (speaker), attack via a man-in-the-middle attack, or compromise the router. This type of attack is unlikely, due to the significant difficulty of performing any of the above tasks.

   This type of attack will lead to similar types of attacks as originating an unauthorized prefix into the peer route table. This type of attack can lead to changing the traffic patterns in an autonomous system, if the prefix is more specific that the original path used to route the traffic.

5. The most dangerous type of attack, in terms of probability, is the Denial of Service attack. There are four methods of attack specified in the attack tree, but as [7] specifies, there are almost an infinite number of attacks and operational conditions which can cause the routing process to stop working as specified, the level of effort required to test all of these is extraordinary. Therefore, special care must be taken to guard against this type of attack, even though the level of effort is much larger than that of other types of attack.

6. The resetting of a single BGP session between two peers is a very likely threat as well. This has already been covered in a previous section, and will therefore not be discussed here.

7. The last type of attack [7] specifies is the sending of a spoofed BGP message. The effects of this type of attack are less well understood. This type of attack is very easy to accomplish, and the attacker could potentially do a great deal of damage with it. This type of attack is used primarily to insert bad or misleading information into the BGP session. It can also be used to reset a BGP session, thus potentially causing a DoS attack. All that is required of the attacker is to either do a TCP Number attack (specified earlier), or to an "intercept and modify" attack (a man-in-the-middle attack) to insert the bad data, and to create a valid BGP packet to insert. This type of attack requires a good amount of skill and processing power, and so this is not necessarily a good choice of attack for a potential attacker. Therefore, this type of attack is not likely to occur, but it is good to know about it in case an attacker chooses this type of attack.

5. SMTP

SMTP, or Simple Mail Transfer Protocol, is, as its acronym states, a simple protocol. Developed in November, 1981 in RFC 788, this protocol was intended to do a particular task: namely, to transfer mail in a reliable, efficient manner. This protocol replaced the already working MTP, or Mail Transfer Protocol (RFC 772, RFC 780). Within the protocol, there exists what has come to be known as an ‘implicit contract’ between the client using the protocol and contacting the server and the host server on which the SMTP daemon is running. The protocol is based on and still mostly uses a simple text based communication between the client and the server.

Even the latest standard of the SMTP protocol, RFC 2821, admits that the protocol in inherently insecure. Section 7 of the RFC 2821 specification relates many of the security problems which plague the protocol. This is not to say that the protocol is flawed; rather for most of the population, who just care to send mail to one another, this is a perfectly good approach. The problem comes with attackers who wish to cause trouble and wish to exploit the security holes in the protocol. In effect, the attackers are exploiting the implicitness of the contract between the client and the server for the attacks. As this paper will show, there are many ways that an attack can be made, and deciding where best to spend the effort to mitigate the risks. Because there are so many vulnerabilities in the SMTP protocol, this paper will concentrate only on a subset of the best known and most exploited ones.

A typical main goal can be stated as 'Run malicious code on SMTP server'. This can be accomplished in any number of ways, but through exploitation of certain key flaws in SMTP, an attacker can gain access to any of a variety of services, shells, and resources on the target computer. It is the administrator’s responsibility to know which flaws exist, which are possible targets, and how best to assign resources and defend against those attacks.

As a sample attack tree, let us first consider the problem of phishing. This security concern, albeit a minor one from the perspective of the network administrator, involves email spoofing. Spoofing is an attack which is relatively harmless, unless combined with another form of attack, e.g. phishing, or trying to obtain a person’s private data from them by purporting to be the legitimate business/person/entity to which the person has already given their private data. Usually, the attacker states that the data at the ‘business’ has been lost, or is corrupted, and is needed again from the user. If the user does not give this information, then some penalty will be involved. The spoofer can use the implicit contract between the client and the host and exploit the fact that the server accepts that the client is giving it correct information about where the mail is coming from in the MAIL FROM portion of the information exchange. After the HELO and acknowledgement from the server (return code 250), the client can send a fake address, which resembles that of the real business he/she is pretending to be. For example:

```
Server: 220 mail.zenprogramming Simple Mail Transfer Service Ready
Client: HELO attacker.nasty.com
Server: 250 mail.zenprogramming.com
```

```
Client: MAIL FROM:<support@ebay.com>
Server: 250 OK
Client: RCPT TO:<sucker@zenprogramming.com>
Server: 250 OK
….
```

As one can see, there is no check to see where the mail truly comes from; the server simply accepts that the mail has come from where it has stated is has come from. Couple this with a fake web page, which is of course embedded in the email message, and the victim has little to no idea that they have been taken advantage of. The attack tree which naturally comes from this scenario looks like this:

```
Goal: Gain user’s private data
OR      1. Send a fraudulent email (“phishing”) (easy, good chance of success)
        AND     1. Spoof the email address of a legitimate company
                2. Setup a fake web page that is a copy of the real web site
                3. Get user to supply personal private information
        2. Hack into user’s computer (difficult, poor outcome)
        AND     1. Guess the user’s password
                2. Gain physical access to the computer
                OR      1. Break into the house
                        2. Bribe repair shop when computer is broken
        3. Rummage through garbage to get personal information (easy, poor outcome)
        ….
```

Naturally, the attacker will choose attack 1, phishing. Not only does the attacker have the best chance of succeeding, but it is also a very easy attack, and many people fall prey to it. In the next section, a sample attack tree will be created which shows the possible attacks that an attacker can use against the protocol.

6. A Sample SMTP Attack Tree

Consider that there are many ways of compromising a computer system. The attack tree from the previous section is useful in finding ways that one can defend against potential attacks of gaining personal data from a user. But the SMTP protocol has other dangers which are more difficult to see. A sample set of attacks [8] can be seen in the attack tree in Figure 1.

```
Goal: Compromise the victim’s system through exploiting SMTP security flaws.
OR      1. Run programs through exploiting the MAIL FROM ‘pipe’ security flaw
        2. Find and use administrative backdoor(s)
        OR      1. Use the DEBUG exploit (sendmail exploit)
                AND     1. Ensure target is a sendmail server
                        2. Send large load of email messages
                        3. Wait for server to get out of sync with itself and trigger DEBUG
                                command
                2. Use the WIZ exploit through Telnet
```

```
            AND   1. Send WIZ command
                  2. Enter no password
      3. Find (scan) users on the system to set up a later attack
      OR    1. Use the EXPN command to search for users
            2. Use the VRFY command to search for users
      4. Relay spam through the system
      OR    1. RCPT TO uucp-style recipient
            2. Specify too many recipients
      5. Exploit a possible “buffer overflow” flaw
      OR    1. Send a very long HELO command of over 1119 characters
            2. Send a very long MAIL FROM command
            3. Send a very long RCPT TO command
            4. Send very long MIME file name
      6. Denial of Service (DoS) attack
            OR    1. A very long HELO command of over 12000 characters
                  2. A malformed MAIL FROM command
                  3. A malformed RCPT TO command
      7. Overwrite files on the server
            OR    1. RCPT TO decode
                  2. RCPT TO uudecode
```

Figure 1: Sample SMTP Attack Tree

1. Firstly, an attacker can run certain programs on the target SMTP server by sending a fraudulent MAIL FROM address which contains a pipe, |, and a string of commands after that. For example,

```
HELO
MAIL FROM: |/usr/ucb/tail|/usr/bin/sh
RCPT TO: root
DATA
From: attacker@example.com
To: victim@example.com
Return-Receipt-To: |foobar
Subject: Sample Exploit
```

This attack exploits the UNIX target system by sending shell executable code hidden inside the email address. Some systems down the line which will handle the email may run this code, and in effect grant access to the attacker. This is, however, a well known attack and is not likely to be easy to accomplish, except on poorly guarded machines. This is most likely handled through patches or upgrades to the sendmail program, and is therefore not a likely target of attack these days.

2. Finding administrative backdoors seems to be a more likely attack strategy. The attack using the DEBUG exploit makes use of slower machines being used to run the sendmail server. If the server receives a large amount of messages, there is a chance that the server can get out of sync with itself, and cause a DEBUG command to be run. This enables an attacker to gain access to the system, because

the DEBUG command allows anyone access to the system as root. But again, these types of attacks are also well known and most likely prevented already.

3. The scanning of users' email addresses is not a direct attack per se, but rather a means to getting a list of email addresses, possibly for spamming. Using either of the commands EXPN of VRFY, an attacker can gain information about the users of that system. This is potentially unavoidable, unless the administrator wishes to disallow clients requesting information about users. This can violate the RFC 2821, and may upset some legitimate users of the commands. As stated before, this is not a direct attack on the system, but rather a means of getting information about users on the system for a possible setup for a later attack.

4. Hijacking an SMTP server for use in spamming is a popular technique. Using an old style RCPT address, which includes a '%' as part of the email address, the spammer can often send unauthorized email messages to recipients, because the software fails to check for those old style formatted email addresses. This is a low risk attack, as there is not a great deal of information or control which can be retrieved as a result of attacking a system this way, but it can still be an annoyance to those who receive the spammer's email messages.

   Additionally, an attacker can 'leech' the connection of an SMTP server by sending a message with a large number of recipients. The server is then sending out the messages instead of each individual email message coming one at a time from the originator. This type of attack also hides from where the spammer originated.

5. The most likely of all the attacks is the dreaded, but well known "buffer overflow" exploit. This attack uses knowledge of the internal memory of a computer and the memory stack to try to hijack a computer. A buffer is an area of storage that is temporarily allocated. Assuming there are no guards against the input to the buffer and the length of a buffer, the attacker can send overflow the buffer by sending it far more data than it can handle. Carefully crafted code can then be run, which takes advantage of knowledge of the size of the variables on the stack, and the location on the stack, which holds the location where this function will return. Once that is known, the attacker can simply redirect the return to his/her malicious code, which will then run. If this is a command to open a command window, this essentially gives the attacker the privileges of the hijacked SMTP server program. This is a well known attack, but often the flaws are not completely known. This is most likely where an attacker will choose to infiltrate, because of the ease of attack and the payoff for the attack is high.

6. An attacker can also choose to try to cause a Denial of Service (DoS) attack. A DoS attack occurs when a server is flooded by excessive amounts of fraudulent requests (HTTP, SMTP, etc.) and tries to reply to all of them. This results in the server being unable to serve the legitimate users. Through sending a very long HELO command, a malformed MAIL FROM, or a malformed RCPT TO command, the attacker can potentially cause the server to crash, which will cause

either a shutdown or a reset of the server. This is a well known attack, and most likely the latest patches and upgrades to the mail server software can prevent against this type of attack.

7. The last type of attack this paper will focus on the overwriting of files on the target SMTP server. Through use embedding the name **decode** or **uudecode** into the RCPT TO command, some unprotected servers will allow the code embedded in the DATA portion of the email message to overwrite a special file. For example,

   ```
   HELO
   MAIL FROM: test@example.com
   RCPT TO: decode
   DATA
   begin 644 /usr/bin/.rhosts
   $*R`K"@``
   `
   end
   .
   QUIT
   ```

   This will insert the text '+ +' into the .rhosts file, which tells the server to trust all who login using the rlogin command (or similar programs) to the server. As with the above, this is a well known attack, and is most likely guarded against on most servers. However, the payoff can be extremely beneficial to the attacker for very low risk and effort expended.

7. Conclusion and Future Work

This paper has described what attack trees are, how they can be used to prioritize resource allocation and defend against potential attacks. Two specific protocols, Border Gateway Protocol (BGP) and the Simple Mail Transfer Protocol (SMTP), were also analyzed, and the various types of attacks were enumerated. While the attack tree used to describe the SMTP vulnerabilities is quite thorough, there are still many other types of vulnerabilities which can be exploited. Additional research and simulated attacking will need to be done in order to discover different means of attacking the SMTP protocol, but it is well worth any administrator's time and effort to develop an attack tree for determining which security threats exist, and which can be guarded.

A. References

[1] Bruce Schneier, Attack Trees. In *Dr. Dobbs Journal*, December 1999 http://www.schneier.com/paper-attacktrees-ddj-ft.html

[2] Unknown, *Attack Trees: It's a Jungle Out There*, 2005 SYS-CON Media, Inc. http://websphere.sys-con.com/read/43842_p.htm

[3] J. Viega and G. McGraw, Risk Analysis: Attack Trees and Other Tricks. In *Software Development*, August 2002. http://sdmagazine.com/documents/s=818/sdm0208a/

[4] Andrew P. Moore, Robert J. Ellison, Richard C. Linger, Attack Modeling for Information Security and Survivability, March 2001
http://www.cert.org/archive/pdf/01tn001.pdf

[5] Sean Convery, David Cook, Matt Franz. BGP Attack Tree
http://www.ietf.org/proceedings/02nov/slides/rpsec-3/rpsec-3.ppt

[6] Jintae Kim, Steven Y. Ko, David M. Nicol, Xenofontas A. Dimitropoulos, George F. Riley, A BGP Attack Against Traffic Engineering. In *Proceedings of the 2004 Winter Simulation Conference*.
http://www.informs-sim.org/wsc04papers/038.pdf

[7] S. Convery, D. Cook, M. Franz, An Attack Tree for the Border Gateway Protocol. Draft-convery-bgpattack-00, 2002.
http://www.rpsec.prg/drafts/draft-convery-bgpattack-00.txt

[8] Unknown, SMTP. In *Internet Security Systems*, 2005
http://www.iss.net/security_center/advice/Exploits/Services/SMTP/default.htm